

CSS 精粹

The CSS Anthology
101 Essential Tips, Tricks & Hacks



[英] Rachel Andrew 著
丁卫颖 李奕 吴戈 译

CSS精粹



[英] Rachel Andrew 著
丁卫颖 李奕 吴戈 译

人民邮电出版社
北京

图书在版编目 (CIP) 数据

CSS 精粹/ (英) 安德鲁 (Andrew, R.) 著: 丁卫颖, 李奕, 吴戈译. —北京: 人民邮电出版社, 2007.10
ISBN 978-7-115-16572-5

I. C… II. ①安… ②丁… ③李… ④吴… III. 主页制作—软件工具, CSS IV. TP393.092

中国版本图书馆 CIP 数据核字 (2007) 第 108384 号

版 权 声 明

The CSS Anthology: 101 Essential Tips, Tricks & Hacks by Rachel Andrew

Copyright ©2004 SitePoint Pty. Ltd. ISBN: 0957921888

All Rights Reserved.

本书由澳大利亚 **SitePoint Pty.** 公司授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式或任何手段复制和传播。

版权所有, 侵权必究。

CSS 精粹

-
- ◆ 著 [英] Rachel Andrew
 - 译 丁卫颖 李奕 吴戈
 - 责任编辑 刘映欣
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
 - 邮编 100061 电子函件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 三河市海波印务有限公司印刷
 - 新华书店总店北京发行所经销
 - ◆ 开本: 800×1000 1/16
 - 印张: 18.75
 - 字数: 402 千字 2007 年 10 月第 1 版
 - 印数: 1—5 000 册 2007 年 10 月河北第 1 次印刷
 - 著作权合同登记号 图字: 01-2006-7029 号

ISBN 978-7-115-16572-5/TP

定价: 39.00 元

读者服务热线: (010) 67132705 印装质量热线: (010) 67129223

内 容 提 要

本书采用问答的形式，为 CSS 使用过程中一些有价值的经典问题提供了精彩的实践解决方案。本书内容包括文本样式、CSS 图像、导航、表格数据、注册表和用户界面、浏览器和设置支持、CSS 定位和布局以及未来相关技术。

本书的目标读者是每一个需要使用 CSS 的 Web 设计人员和开发人员。本书通过经典的问题和精彩的解答将理论融于实践，使每一个带着问题阅读本书的读者都能找到自己满意的答案。书中所有源代码可在网址 <http://www.ptpress.com.cn> 的下载区免费下载。

关于作者

Rachel Andrew 是 Web 解决方法提供者 edgeofmyseat.com 的 Web 开发人员和主管。在不编写代码的时候，她撰写关于编写代码的书，她是几本关于推广 Web 标准的实践应用以及日常工具和技术之书的合作者。Rachel 有一个观点，即真实世界与 Web 标准是相通的。而她的写作和教学是以每天她在自己的公司里所掌握的经验与这一观点为基础的。

Rachel 和她的丈夫 Drew 以及女儿 Bethany 生活在英国。在工作之余，他们经常会徜徉在英国的乡间，寻找能提供一顿周末午餐和美味啤酒的野外小店和幽雅的小客栈。

关于技术主管

作为 SitePoint 公司的技术主管，Kevin Yank 负责审查全部技术出版物——书籍、文章、新闻信件和博客。他已经为 SitePoint 写了 50 余篇技术文章，所涉及的技术包括 PHP、XML、ASP.NET、Java、JavaScript 和 CSS，但是他最有名的书是 *Build Your Own Database Driven Website Using PHP & MySQL*，也是由 SitePoint 公司出版的。Kevin 现居住在澳大利亚的墨尔本。在闲暇时间他喜欢驾驶轻型飞机和学习即兴表演。

前言

在我还没有写书时，我一直在编写代码。我从事的是创建网站和应用程序的工作，我确信本书的很多读者也在从事我这样的工作。每天，我用 CSS 来完成工作。我知道当一个项目马上要交付时努力工作是什么感觉。

那些不用 CSS 或只将 CSS 用于简单文本样式的设计人员和开发人员，当我和他们交谈时，我反复听到的一件事情是，他们没有时间去学习这种全新的做事情的方法。毕竟，表格和 spacer GIF 还能用，可以用它们完成工作和赚钱。

我是幸运的。我很早就遇到了 CSS，并且出于兴趣开始接触它。后来我的知识随着 CSS 技术的自身发展而增长，现在我每处理一个项目都可以利用我 3 年的经验来建立 CSS 布局。

一些技巧和技术让我能够快速而且容易地用 CSS 开发网站和应用程序。我想通过本书将我的这些技巧和技术介绍给读者。

本书没有大篇幅的理论。读者们所能找到的就是能让他们现在做出很酷的东西的一些解决方法，但是这些方法应该也只是作为自我创造力的起点。凭我的经验，实践比苦读更容易学到东西，所以当读者能够用本书找到解决方法来帮助他在截止日期前建立用户网站并使之运行的时候，请用这些例子做实验并将它们作为学习新技术的一种方法。

本书就是要让读者能够及时地在任何给出的知识点上快速地找到他正在努力思考的 CSS 问题的答案。不需要通读全书，只需抓住需要或感兴趣的技术，然后开始去做就行了。伴随着每个解决方法，我都提供了一个解释来帮助读者理解使用该方法的原因。这个知识将使读者在属于自己的时间里可以拓展和实践该技术。

我希望读者喜欢这本书！写这本书带给我巨大的快乐，同时我希望它将会是一本有用的日常参考书，并且作为一个工具帮助给予读者开发新的 CSS 技术的自信。

本书的读者对象

本书的目标读者是需要使用 CSS 的人们，即那些已经看到了展示出来的很酷的 CSS 设计，但是又没有时间为创建一个站点而费力地研读大量的理论并进行讨论的 Web 设计人员和开发人员。每一个问题都给出了一个可执行的解决方案，可以执行该方案来解决问题或者以此作为起点继续深入下去。

本书不是一本指导书。第一章包含了 CSS 最基础的知识，并且前面章节包含的技术要比后面章节包含的简单；如果读者已经有一定的 CSS 基础，那么就会发现掌握例子会容易些。

本书包含的内容

第 1 章 CSS 入门

本章没有遵循其他章节的安排方法，对于任何需要复习 CSS 基础知识的读者来说，它只是一个快速的 CSS 指导。如果读者已经在自己的项目中使用着 CSS，那么可以跳过本章；如果读者发现需要查找一些基本概念的更多细节，也有可能回过头来参考本章。

第 2 章 文本样式和其他基础知识

本章介绍了为文档中的文本设定样式和格式的技术，随着内容的深入解释了如何设置字号、颜色以及把包围网页元素的恼人的额外白色空白删除。即使读者已经在用 CSS 设定了文本样式，在这里也会找到一些有用的小技巧。

第 3 章 CSS 和图像

将 CSS 和图像结合起来可以创造出震撼的视觉效果。本章介绍了可以做到这一点的方法，并且包含了背景图像（不只是主体部分），还介绍了如何在其他主题中间放置带图像的文本。

第 4 章 导航

我们都需要导航，而本章解释了如何作出 CSS 样式的导航。在本章里涉及的内容有对基于图像的导航用 CSS 替换的问题、CSS “标签” 导航、把背景图像和 CSS 文本结合起来创造出吸引人的而且易用的菜单以及以本章介绍的一种易用的方法用列表来构建导航。

第 5 章 表列数据

虽然在任何地方都应该尽可能避免用表格来布局，但是表格应该体现它的真实意图：显示表列数据，比如包含在一个电子数据表里的数据。本章将介绍一些技术，它们通过应用表格来创造出有吸引力的而且可用的表列数据显示。

第 6 章 注册表和用户界面

作为一个设计人员或开发人员，很可能要花相当长的时间来创建用于数据录入的注册表。CSS 可以帮助读者创建出更具吸引力和更好用的注册表。本章展示了在考虑到关键的易用性原则的前提下怎样才能做出吸引人的注册表。

第 7 章 浏览器和设备支持

怎样处理有 CSS 故障的旧式浏览器以及怎样更改设备？这些问题构成了本章的主题，本章介绍了如何排除 CSS 故障以及去哪里寻求帮助，还讨论了在尽可能多的浏览器上测试站点的方法。

第 8 章 CSS 定位和布局

本章探究了用 CSS 来创造出漂亮的而且易用的网页的技巧。其中概括了一系列不同的

CSS 布局和各种技术，可以把这些结合起来并且在此基础上扩展，从而创造出一系列不同的网页布局。

第9章 实验、浏览器专用 CSS 和未来技术

通篇至此，已经介绍了可以在所有浏览器上运行良好的技术，并且为了运行的需要没有包含进大量的 CSS “黑客”。但是，在本章中也介绍了更加新颖的技术，它们不能够在所有浏览器上运行得很好，或者为了运行良好还需要做相当多的额外工作。

致谢

首先，我要感谢 SitePoint 团队，是他们和我融洽地进行沟通并使这本书付诸出版，尽管事实上我们有各自不同的时间段，每天他们开始工作的时候却看到我要去睡觉了。尤其必须要感谢 Simon Mackie，在整个写作过程中他的鼓励是对我巨大的支持。

还要感谢技术主管 Simon Willison，是他挑出了最细微的错误，他对细节的关注和深思熟虑的意见保证了本书的精确和清晰。

在此，对那些在 CSS 世界里确实开辟出一片新天地的人们，对那些在本书中讨论过他们的想法的人们，还要对那些与大家分享自己的想法和创造力的人们说一声“谢谢。”

感谢 Drew 对我的支持和鼓励，感谢他在我做出本书的例子时乐于同我讨论 CSS 概念，感谢他在我苦恼时逗我开心，并且感谢他容忍我们某些社会生活的完全空白。最后，必须感谢我的女儿 Bethany，她非常理解她的妈妈不停地在电脑前工作，而且是她提醒我什么才是每天里重要的事情。是你们两个让这么多的事情成为可能，感谢你们。

目 录

第 1 章 CSS 入门	1
1.1 HTML 存在的问题	1
1.2 用 CSS 定义样式	2
1.3 CSS 选择符	5
1.4 小结	8
第 2 章 文本样式和其他基础知识	9
技巧 1 如何用 CSS 替换字体标记	9
技巧 2 可否使用 pixel、point、em 或其他的单位设置字号	10
技巧 3 如何指定文本以特定的字号显示	15
技巧 4 如何删除链接的下划线	16
技巧 5 如何创建当鼠标滑过时改变颜色的链接	17
技巧 6 如何在页面上显示两种不同样式的链接	19
技巧 7 如何为标题添加背景色	20
技巧 8 如何定义带下划线的标题样式	21
技巧 9 如何取消<h1>标记与其下段落之间的较大间隙	22
技巧 10 如何不用字体标记就能突出显示页面中的文本	23
技巧 11 如何改变文本中的行距	24
技巧 12 如何水平对齐文本	25
技巧 13 如何定义水平线的样式	26
技巧 14 如何缩进文本	27
技巧 15 如何使文本居中	28
技巧 16 如何用 CSS 使文本全部变为大写	29
技巧 17 如何改变或删除列表项的项目符号	30
技巧 18 如何用图像作为列表项的项目符号	32
技巧 19 如何在目录中取消缩进的左边距	32
技巧 20 如何水平显示列表	33

技巧 21	如何为 CSS 文件添加注释	34
技巧 22	如何不在主体标记中添加属性就能取消页边距	35
小结		35
第 3 章 CSS 和图像		36
技巧 23	如何为图像加边框	36
技巧 24	如何用 CSS 替换图像上不支持的 HTML 边框属性	38
技巧 25	如何用 CSS 为网页设置背景图像	38
技巧 26	如何定位背景图像	40
技巧 27	在文本随着网页的滚动而移动的时候, 如何使背景图像保持不动	42
技巧 28	如何为其他元素设计背景图像	43
技巧 29	如何将文本放置在一幅图像的上部	46
技巧 30	如何向文档里添加一幅以上的背景图像	47
小结		48
第 4 章 导航		49
技巧 31	如何用 CSS 替换图像导航	49
技巧 32	如何像设置导航菜单一样设置结构列表的样式	54
技巧 33	如何在不使用图像和 JavaScript 的情况下用 CSS 创建翻转导航效果	57
技巧 34	能否用 CSS 和目录创建包含子导航的导航系统	58
技巧 35	如何用 CSS 和列表创建水平菜单	62
技巧 36	如何用 CSS 创建按钮式的导航	65
技巧 37	如何用 CSS 创建标签导航	67
技巧 38	如何改变光标样式	73
技巧 39	如何在 CSS 中不使用 JavaScript 创建翻转效果	75
小结		78
第 5 章 列表数据		79
技巧 40	如何用 CSS 布局电子数据表中的数据	80
技巧 41	如何保证自己的列表数据既可用又吸引人	81
技巧 42	不用 HTML border 属性如何为表格添加边框	83
技巧 43	在已经用 CSS 添加了边框的情况下, 如何阻止表格的各个单元间 出现空白	85
技巧 44	如何用一种吸引人的而且可行的方法来显示电子数据表数据	86
技巧 45	如何用交替的颜色显示表格行	89

技巧 46	如何在鼠标移动到某一表格行的上面时改变其背景颜色	92
技巧 47	如何用 CSS 显示日历	94
小结		104
第 6 章 表单和用户界面		105
技巧 48	如何使用 CSS 设计表单上的元素	106
技巧 49	如何对一个表单中不同的域应用不同的样式	109
技巧 50	如何避免表单上出现多余的空白和裂痕	111
技巧 51	如何使提交按钮看起来像文本	112
技巧 52	当用户使用的设备只能浏览文本时, 如何确保这些用户能够理解完成 表单的步骤	112
技巧 53	如何用 CSS 代替表格设计一个含有两列的表单	115
技巧 54	如何将有关联的域组成一组	119
技巧 55	如何设计快捷键的提示字母	123
技巧 56	如何在选择菜单中使用不同的颜色来强调各菜单项	125
技巧 57	用户往表单上输入数据时就像往电子表格中输入一样, 用 CSS 如何 设计这个样式	126
技巧 58	当用户点击时如何突出表单域	132
小结		133
第 7 章 浏览器和设备支持		134
技巧 59	应该在哪些浏览器中测试网站	135
技巧 60	当只有一个操作系统可以使用时, 如何在很多浏览器上测试	136
技巧 61	是否有一种服务能够帮助用户在各种浏览器中查看网站的运行情况	138
技巧 62	能否在 Windows 系统中安装多个版本的 IE 浏览器	139
技巧 63	如何在纯文本浏览器中测试网站	141
技巧 64	如何在屏幕阅读器中测试网站	142
技巧 65	如何在 Netscape 4 中隐藏 CSS	143
技巧 66	如何在 Netscape 4 中显示各种不同的样式	144
技巧 67	如何只在第 4 版浏览器中增加一个消息, 以解释为什么网站 看起来很普通	148
技巧 68	在其他浏览器中如何隐藏 CSS	150
技巧 69	为什么网站在 IE 6 和在 Mozilla 中的显示效果不一样	154
技巧 70	发现了一个 CSS 问题, 怎么办	157
技巧 71	在 IE 6 中一些内容有时出现有时消失, 该怎么办	159

技巧 72	在 W3C Validator 中的错误和警告信息是什么意思	163
技巧 73	如何给一些特殊的设备（如屏幕阅读器或网络电视）创建样式表	164
技巧 74	如何创建打印样式表	166
技巧 75	一些浏览器允许用户选择样式表，如何向网站添加可选择的样式表	173
技巧 76	如何制作样式表转换器	176
技巧 77	如果不复制代码如何使用可选择样式表	179
小结		183
第 8 章	CSS 定位和布局	184
技巧 78	如何决定什么时候使用 class，什么时候使用 ID	184
技巧 79	能否将一个内联元素显示成一个块级元素，反之是否可行	185
技巧 80	在 CSS 中，边界和补白如何设置	187
技巧 81	如何不使用 HTML 对齐属性使文本环绕在图片周围	190
技巧 82	当设置浮动属性时如何防止下一元素上移	192
技巧 83	不使用表格如何左右对齐 logo 和副标题	196
技巧 84	如何使用 CSS 在页面上对一个对象定位	199
技巧 85	如何使块在页面上居中	203
技巧 86	如何创建一个动态的、菜单在左边而内容在右边的双栏布局	205
技巧 87	能否反转布局将菜单显示在右侧	211
技巧 88	如何创建一个固定宽度、居中且有两栏的布局	212
技巧 89	如何创建一个三栏的 CSS 布局	222
技巧 90	如何用 CSS 添加正常显示的页脚	233
技巧 91	如果不使用表格如何显示缩略图库	239
小结		243
第 9 章	实验、浏览器专用 CSS 和未来技术	244
技巧 92	如何建立彩色滚动条	244
技巧 93	如何创建一个菜单，当网页在下面滚动时它保持不动	246
技巧 94	如何让固定的菜单在 IE 里工作	250
技巧 95	能否用 CSS 创建保持在原位固定不动的网页脚注，使之看起来如同框架一样	254
技巧 96	能否创建纯 CSS 的下拉菜单	260
技巧 97	能否在 CSS 边框上创造出圆角	266
技巧 98	如何在所有的浏览器里用 CSS 制作出圆角	268
技巧 99	如何让元素在基于 Mozilla 的浏览器和 IE 里都是半透明的	273

技巧 100	如何用 CSS 向访问者指明哪些是外部链接	276
技巧 101	能否用 CSS 把文本嵌入文档里	278
技巧 102	如何设定模块的第一行或者第一个字母的样式	279
技巧 103	使用在某些浏览器中不能工作的效果是否是一件坏事	283
小结		283

第 1 章

CSS 入门

层叠样式表听起来很吓人。单是这个名字就让人联想到神秘的代码和语法，使初学者不得要领。但是，实际上对网页开发人员来说 CSS 是一种最简单也最方便的工具。第 1 章采取了与本书其他各章不同的介绍方式。在这一章里将指导读者通览 CSS 的基础知识，向读者展示如何使用 CSS 来简化管理一个格式统一的网站。如果读者已经在自己的网站上用 CSS 来格式文本，可以跳过本章直接从第 2 章的技巧开始阅读。

1.1 HTML 存在的问题

CSS 是一种语言，用来定义应用于网站中的格式，包括颜色、背景图像、字体、外边距和缩进。如果以前从未用过 CSS，那么可以原谅这种想法，即“用 HTML 标记就能做那些事情，为什么需要 CSS 呢？”对这个问题最好的回答就是，举例说明用 HTML 定义样式时产生的问题。

现在，一个流行的设计选择是对网站的主体文本使用 sans-serif 字体（例如 Arial、Verdana、Tahoma 等）。因为多数的网页浏览器默认一种 serif 字体，比如 Times New Roman，所以用一种 sans-serif 字体来创建一个复杂的网页设计通常会包含很多的〈font〉标记。在一个复杂的设计里，可能会看到针对一个网页上的所有文本，会有十多个〈font〉标记专门用于指明同

一字体。将这个数字乘以 5——一个最简约的网站的网页数——就会与上百个标记为邻了。一个大型网站可能有 50 个甚至更多的网页，在这种情况下将看到上千的〈font〉标记，所有这些标记都只应用于文档文本中的一个基本的、一致的样式。

现在麻烦来了。在一个星期五下午，你的客户给你打电话说：“Verdana 字体是不错，但是每个人都用它。让我们换用 Tahoma 吧。”抛开特殊的搜索替换工具不论，你现在面临的任务是调整 100、1000 乃至更多的〈font〉标记来完成这一改变，从客户的角度来看这似乎是一个很简单的改动。你可以向早已计划好的周末滑雪告别了。并且，最好不要发出大声的叹息，大多数用户对此是不会留有好印象的。

如果了解 HTML，可能会想到〈basefont〉标记，在整个网页上设置一种默认字体，这样可以为该问题提供一个很好的解决方法。但是即使这样，仍需为网站的每一个网页调整一个标记。同样地，要添加另一种字体样式（如果想为漂亮的导航栏使用一种不同的字体），问题就全来了。

不使用 HTML 来格式化网站的另一原因是，这些呈现出的元素是禁用的（标记出来，在以后的规格里将被删除），如果想使用 Strict DOCTYPE（比如 HTML 4.01 Strict 或者 XHTML 1.0 Strict），就不能用这些元素。如果使用一个改进的 DOCTYPE，网页将是有效的，但是在可能的地方最好还是避免使用这些不提倡使用的元素。通过本书里的例子你将会发现，CSS 允许你做很多只用 HTML 无法完成的事情。所以，有很多理由选择使用 CSS。下面，让我们停止讨论来看一些 CSS 的实际应用吧！

1.2 用 CSS 定义样式

CSS 的基本意图是允许设计人员定义一个样式（一系列的格式细节，如字体、字号和颜色），然后用一个选择符将 CSS 应用在一个或多个 HTML 网页的一个或多个部分里。下面通过一个基本的例子来了解这是怎么做到的。

思考下面 HTML 文档的框架：

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Simple Page</title>
<meta http-equiv="content-type"
    content="text/html; charset=iso-8859-1" />
</head>
<body>
<h1><font face="sans-serif" color="#3366CC">First Title</font>
</h1>
<p>...</p>
```

```

<h2><font face="sans-serif" color="#3366CC">Second Title</font>
</h2>
<p>...</p>
<h2><font face="sans-serif" color="#3366CC">Third Title</font>
</h2>
<p>...</p>
</body>
</html>

```

这个文档包含 3 个用〈h1〉和〈h2〉标记创建的标题。为了让这些标题更显眼，使用了〈font〉标记使它们以淡蓝色、sans-serif 字体（例如 Windows 浏览器会用 Arial 字体显示它们）来显示。注意，甚至在这样初级水平上也有相关的重复。要得到想要的字体细节，不得不独立地分 3 次指定。难道只定义一次字体然后将其应用在标题上没有意义吗？这是 CSS 版本：

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Simple Page</title>
<meta http-equiv="content-type"
content="text/html; charset=iso-8859-1" />
<style type="text/css">
h1, h2 {
font-family: sans-serif;
color: #3366CC;
}
</style>
</head>
<body>
<h1>First Title</h1>
<p>...</p>

<h2>Second Title</h2>
<p>...</p>

<h2>Third Title</h2>
<p>...</p>
</body>
</html>

```

所有的技巧都存在于文档的〈head〉里的〈style〉标记中间，在那里定义了淡蓝色的 sans-serif 字体并将该定义应用在文档里所有的〈h1〉和〈h2〉标记上。不用担心语法，稍后将详细解释。同时，〈font〉标记已经从〈body〉里完全消失了，使文档看上去整齐多了。对网页顶端的样式定义的改动不仅将影响这 3 个标题，还将影响添加在网页上的其他标题。

现在，对 CSS 是做什么的有一个印象了，接下来解释一下在 HTML 文档中使用 CSS 样式的不同方法。将 CSS 样式嵌入网页最简单的方法就是用〈style〉标记，如同在前面例子里做过的那样。该方法可以通过将 CSS 样式嵌入在〈style〉标记里来声明任意数量的 CSS 样式，

如下所示:

```
<style type="text/css">
CSS Styles here
</style>
```

`type` 属性指定了定义样式所要用的语言。CSS 是唯一广泛应用于定义样式的语言,用 `text/css` 值来表示它。

虽然 `<style>` 标记既好用又简单,但是它有一个大的缺点。具体地说,如果想在整个网站都使用特殊的一套样式,将不得不在网站的每一个网页顶端的 `<style>` 标记里重复那些样式定义。

一个更明智的选择是将这些定义放入一个纯文本文件里(通常文件名为 `.css`),然后将文档链接到那个文件。在那个文件里,任何样式定义的改动都将影响链接到该文件的所有网页。这样就达到了先前提到过的全网站范围内的样式定义的目的。

为了将一个文档链接到一个 CSS 文本文件(例如, `style.css`),可以在文档的开始嵌入一个 `<link>` 标记:

```
<link rel="stylesheet" type="text/css" href="styles.css" />
```

回到最初那个 3 个标题共用一个样式的例子,现在那个文档看起来将会像这样:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>A Simple Page</title>
<meta http-equiv="content-type"
content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" type="text/css" href="styles.css" />
</head>
<body>
<h1>First Title</h1>
<p>...</p>

<h2>Second Title</h2>
<p>...</p>

<h2>Third Title</h2>
<p>...</p>
</body>
</html>
```

`style.css` 文件将包含该样式定义:

```
h1, h2 {
font-family: sans-serif;
color: #3366CC;
}
```

如同用一个图像文件一样,可以在需要的网页里重复使用这个 `style.css` 文件。这样不仅可以节省输入量,还可以保证贯穿整个网站的标题有一致的外观。